Maximizing Social Media Influence by Evolving Kickoff Seeds under Constraint Circumstances

Md Abdur Rahaman
Dept. of Computer Science
University of New Mexico
abdrahaman005@unm.edu

Sarika Kumar
Dept. of Computer Science
University of New Mexico
kumars@unm.edu

Farhan Asif Chowdhury
Dept. of Computer Science
University of New Mexico
fasifchowdhury@unm.edu

***Abstract*—A social network plays a fundamental role as a medium for spreading information, views and influence across the people. Apparently, for different political and non-political purposes (e.g., business, volunteering etc.) leveraging that influence over a giant social network is of great importance. In influence maximization problem the objective is to find a small subset of users who can maximize the spread of influence over the network. This influence maximization problem is a NP-hard problem, which implies an exhaustive search algorithm will not provide a solution under feasible run-time and greedy algorithms require heuristic which imposes several limitations. In this paper, we propose a method using genetic algorithm (GA) to solve the influence maximization problem. Using a real world social network dataset, we evaluate the performance of our method in comparison to greedy and random selection algorithm. Experimental results show that our method outperforms greedy and random selection algorithm.**

***Index Terms*—Social Network, Genetic Algorithm, Influence Maximization Problem, Elitism, Mutation, Crossover**

I. Introduction

In today's highly connected world, social media plays a vital role in many aspects of our daily life. Our daily choice's and opinions gets influenced by the social interactions and the activities of our social media friends. These interactions and relationships can be represented using graphs where nodes represent the users and the edges represent the relationship between them. The identification of influential users in social networks is important. At first, it is not very obvious that identification of influenced users are helpful but it has many subtle applications. For example, during natural disaster, there is an urgency that aid information needs to be dispersed as quickly as possible. When a company is launching a new product, they are looking for those early adopters who can spread the information about the new product to a greater number of audience via *word of mouth* marketing.

Identification of a small subset of users in a network is called the influence maximization problem. To understand social influence better, Kempe et al. represented the whole social network as a graph G where each node v represents a user in the social network, each edge represents a relationship between the user and the weight represents the degree of closeness of a relationship. Kempe et al. [1] demonstrated that both diffusion models are NP-hard problem, which implies that an exhaustive search algorithm will not provide a solution under feasible run-time. This problem is NP-hard because of the necessity of repeatedly calculating the influence of every node.

Since influence maximization problem is a complex problem, there have been many proposed solutions. Greedy algorithm requires heuristic which imposes several limitations. In this paper, we propose a solution to this problem using genetic algorithm. We perform experimental evaluation using real social network dataset to compare the performance of our proposed method against greedy and random selection methods.

II. Problem Statement

A social network of V number of nodes (i.e users) and E number of edges (i.e connection between users) are given where each node has a specific number of connections and each connection propagates information based on probabilistic value using a diffusion model. From V users, we want to identify a subset of k initial users who can propagate the information to larger audience within a given time frame.

A. Formal Definitions

- *Nodes*: A node is the fundamental unit of which graphs are formed. In our context, nodes are the user accounts.
- *Edge/Connection*: In this context, an edge represents a relation (friend for instance) between two user accounts. The weight of the connection is a probability value which corresponds to the rate of information this channel/connection exchange back and forth.
- *Reachability*: In graph theory, it refers to the ability to traverse from one vertex to another. In our context, it refers to the ability of one user to connect to other user's within a given time frame. The connection could be a direct relation between the source and the destination or they are connected through series of other users.
- *Social Influence:* It is a measure of reachability of a user in term of number of users it is directly or indirectly connected to.
- *Seed*: An initial user (starting node) who receives a information first.
- *Chromosome* : A subset of 'L' randomly selected seeds where L is a parameter of the method.

III. Related works

The influence maximization problem was first formalized by Kempe et al. [1]. Their objective was to find a seed set A of k nodes such that the number of the influence nodes from seed set A is as large as possible. There were two representative diffusion models proposed to find the users who have maximum influence over the network: the linear threshold model and the independent cascade model. The independent cascade model simulated the "word of mouth" of real world where a weight was added between the two nodes to know whether one user is influenced by the other friend or not. Kempe et al. proved that the IM (Influence Maximization) problem is an NP-hard problem and proposed a basic greedy algorithm called KGA to solve this.

Greedy algorithms [1] [2] have been extensively used to solve the influence maximization problem. In general in a greedy algorithm, based on a specific heuristic a single new node that maximize the influence is added to an initially empty set of seed in each iteration until it reaches the maximum allowed seed number. In [3], the authors used Simulated Annealing method which has lesser complexity than greedy algorithm but doesn't guarantee any optimality. There have been a few recent work [4] [5] [6], where GA have been used to find an optimal solution.

IV. Proposed Method

We propose a genetic algorithm based influence maximization algorithm where we use GA to evolve different randomly generated initial seed sets over the generations. Our GA has a well-defined fitness function, selection of best performing elites, mutation and crossover. In the first step, we randomly generate an initial population of N seed sets. For each individual member of the seed sets, we traverse across the given network to compute the fitness value. We define the fitness value as the number of nodes or users one particular seed set can reach out under a given time frame. Based on the fitness score, we select the top elite seed sets and on that selected set of elites we perform mutation and crossover to shift through the generations. We run this GA evolution procedure for a specified number of generations and in the end we obtain a set of most influential seeds for the given network.

We represent the network in an adjacency matrix format. If there is an edge between node i and j, we put a 1 in the (i, j) position of the adjacency matrix. We use *Independent Cascade Model* as the diffusion propagation model with a probability value $p = 0.5$, each edge has 50% chance of propagating any information from one user to another. We have associated several randomly generated probabilistic values with each edge connection between 0 to 1. Based on that probability value, on each case an edge will either propagate the message or not. In brief, we create several random version of our network, and in each version an edge is either active or inactive with equal probability.

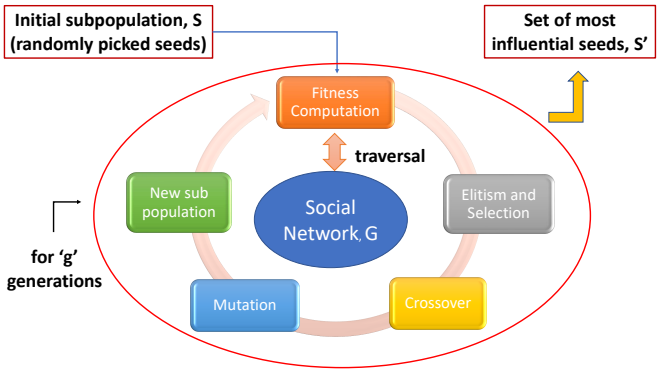

Fig. 1: A flowchart diagram describing the overall methodology of the proposed GA based algortihm.

TABLE I: Genetic Algorithm Parameters

Parameter	Description
V	Number of nodes in the network
E	Number of connections
L	Length of the chromosome or number of users in a seed set
N	Number of chromosomes in an initial population
g	Number of generations
T	Time steps
Mu	Probability of mutation
Co	Probability of crossover
O	% of overlap between two chromosomes of a subpopulation

Initialization of the seed sets: We prepare a set of all user ids from the network and assign a numeric serial number to all ids. We use the assigned numeric serial key ids to generate N number of seed sets with a length of L. During the initial seed set generation, we keep each member of one particular seed set unique, which implies that there is no duplicate user or node in one seed set. **Fitness Function:** Our main objective is to maximize the influence over a social network. Essentially, our fitness function accounts for the that influence of an user; which is a measure of how many other unique users one user can reach within a given time frame. So, for one seed set, we compute the fitness by calculating the reachability of all the member of that seed set as a whole, an overall count of number of unique users the traversal algorithm can possibly visit if it is starting from that set of initial seeds. We use a modified depth-first-search (DFS) algorithm to traverse through the network to calculate the spread of the users of a seed set.

Selection and Elitism: We select a percentage of highest fitness individuals as elites from each generation. Then, we choose two random elites from all the selected elites to perform crossover and mutation.

Crossover: Here, we select a random position and slice two subsets of seeds at that position. Then, we exchange the ending part of both subsets between each other. That produces two different subsets of seeds.

Mutation: We select a random bit again from both of the newly generated chromosomes and then we flip that bit to

engender two new chromosomes. These new seed sets are going to be added to the subpopulation for the next generation. Before adding the new members in that seed sets, the method ensures it does not overlap with any existing member. In each iteration, the evolution creates two new seed sets and continues the process until generates a brand new subpopulation.

in figure 1, we show an overall methodology flowchart of our proposed GA-based influence maximization algorithm.

V. Experimental Results

We evaluate the performance of our proposed GA-based Influence Maximization Algorithm using a real world network. We have compared our proposed method's performance against greedy and random selection algorithm.

A. Dataset

We have collected one real-world large social network datasets from SNAP repository [7]. We have used the WikiVote dataset, which is compiled from the public voting information of Wikipedia admin election process that happened in 2008. In this dataset, each node represents one Wikipedia user and each edge connection represents an endorsement between two users for Wikipedia admin-ship. In Table II, we report few descriptive statistics of the used dataset.

TABLE II: Wiki-Vote Dataset Statistics

Statistic	Value
Number of nodes in the network	7115
Number of edges in the network	103689
Average Degree	27

B. Performance of GA over generations:

We have used *Independent Cascade Model*, with an independent propagation probability fixed over the network as $p = 0.5\%$. This 50% propagation probability implies that, at any time each edge between two users have 50% chance of propagating any news received by one user to the other user. We have associated twenty randomly generated probabilistic values with each connection between 0 to 1. Based on that probability value, on each case an edge will either pass a message or not. In brief, we create twenty random version of our network, and in each version one edge is either active or inactive with equal probability. For each seed set, we calculate the fitness value on each of these twenty randomized version and we finally take average over all these twenty fitness values to calculate mean fitness value.

We run our genetic algorithm to evolve for 100 generations. Initially, we randomly generate 100 seed sets. Then on each generation, we calculate the fitness value for each seed to select 10 best performing seed set as elites. Afterwards, we perform crossover and mutation on these elites to create 90 more seed set. We use these 100 seed sets for fitness value calculation in the next generation. In Table III, we describe the parameters of our used GA algorithm.

We have varied the length of the seed set to measure the robustness of our GA algorithm. We have used seed set length

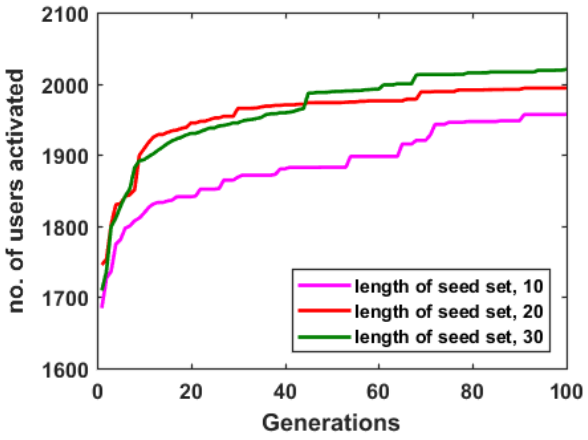

Fig. 2: GA fitness values over generations for different seed set length.

10,20 and 30. In figure 2, we show the change in fitness value which is represented as the number of users activated over generations for all three seed set lengths. We notice that, for all three seed set length, the initial fitness value started from lower value and increased over generation to reach a plateau after around 80 generation. We observe that, even in that plateau region, there are few slight increases. Due to computational resources we could not run our GA algorithm for higher number of generations. We hypothesize that, we would observe more increment in fitness value in the subsequent higher generations.

TABLE III: GA Parameter Values

Parameter	Value
Number of generations	100
Initial population size	100
Seed set length	$10, 20, 30$
Number of Elites	10% of population
Mutation rate	10% of seed set size

C. Performance Comparison

We compare the performance of our GA-based influence maximization algorithm with greedy and random selection algorithm. In greedy selection algorithm, the seed set members are selected based on the individual influence of each users. The users who have highest influence over the network are included in the seed set. In random selection algorithm, we randomly generate some seed sets, and select the one with highest influence. In our experiment, we have randomly generated 10,000 seed set to select the best performing seed set.

In figure 3, we display the performance of the GA, greedy and random selection algorithm for three different seed set length. We observe that, in all three cases GA outperformed both greedy and random selection algorithm for this particular dataset. As expected, the maximum influence has increased as we increase the seed set length for both GA and greedy

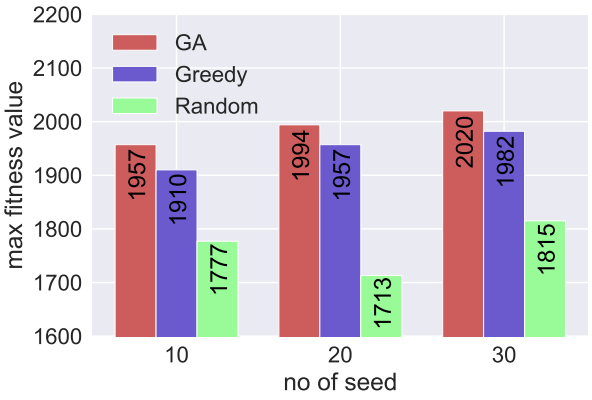

Fig. 3: Influence maximization performance comparison between GA, Greedy and Random selection algorithm.

algorithm. But, due to the randomness associated with the random selection algorithm, for seed set length 20 the maximum influence is less than the maximum influence for seed set length 10.

D. Runtime

We have used *Matlab* for the implementation of all three algorithms. We run our algorithm on a machine of 2.7 GHz Intel Xeon processor with 88 cores and 1TB RAM. Our greedy algorithm took approximate 1 Hour to finish. We configured our GA to run for 100 generation which took approximately 50 minutes. The random selection algorithm finished in approximately 30 Minutes.

VI. Discussion

There are few limitations of our method. The most important one is it is highly biased to the initial population set and it doesn't always ensure the best solution of a problem because of the random probabilistic assumptions it is making while traversing through the network. Here, we are trying to model human behaviour for a given situation which is a hard scenario to approximate. For some given queries, the number of node is less important than quality of nodes (nodes of specific interest). For instance, valued customers of an organization. Our current method can't implement those complicated social interactions.

VII. Future Work

We have compared the performance of our GA algorithm against naive greedy algorithm. In future, we would like to compare our algorithm against other heuristic based greedy algorithm. We would optimize the GA parameters to increase the performance of the GA-based method. We want to evaluate the robustness of our proposed algorithm on other networks with different network properties. We would like to observe how our proposed GA-based algorithm performs in networks that are dynamically modified based on Barabasi preferential attachment [8] as influence changes over time.

VIII. Conclusion

In this paper, we have used Genetic Algorithm to solve the influence maximization problem. We have evaluated the performance of our proposed method using a real world network dataset against greedy and random selection algorithm. experimental evaluation shows that our proposed method can perform on par with the greedy algorithm under a feasible runtime. Moreover, GA-based algorithm does not require any prior information about the overall network structure. The proposed GA-based algorithm can be further improved by optimizing the GA-parameters, as well as by introducing heuristic in the several GA evolution steps. As the importance and pervasiveness of social networks is on the rise over time, further improvement of the proposed GA-based influence maximization in social networks would open up new applications in other domains of social networks.

IX. Contributions

We (Farhan, Abdur and Sarika) have worked together on this project during discussion, coding, figure generation, write-up using overleaf and proof reading. In particular, Farhan wrote the code for "greedy random selection algorithm, and main function for GA; Abdur wrote code for "main GA algorithm, mutation and crossover"; and Sarika wrote code for "mutation and crossover". Farhan wrote the experimental results, discussion, future work and conclusion. Sarika wrote the introduction and related works. Abdur wrote the abstract and methods. Farhan and Abdur generated all the figures from the experimental results. Farhan formatted the code finally and wrote the readme file. All three members participated in final proof-reading.

References

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," pp. 137–146, 2003.

[2] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 199–208.

[3] Q. Jiang, G. Song, C. Gao, Y. Wang, W. Si, and K. Xie, "Simulated annealing based influence maximization in social networks," in *Twenty-fifth AAAI conference on artificial intelligence*, 2011.

[4] K. Zhang, H. Du, and M. W. Feldman, "Maximizing influence in a social network: Improved results using a genetic algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 478, pp. 20–30, 2017.

[5] C.-W. Tsai, Y.-C. Yang, and M.-C. Chiang, "A genetic newgreedy algorithm for influence maximization in social network," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 2549–2554.

[6] D. Bucur and G. Iacca, "Influence maximization in social networks with genetic algorithms," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2016, pp. 379–392.

[7] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[8] H. Jeong, Z. Néda, and A.-L. Barabási, "Measuring preferential attachment in evolving networks," *EPL (Europhysics Letters)*, vol. 61, no. 4, p. 567, 2003.